Due Date: February 10, 2009

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	)	
	)	
Inventor: Paul J. McArdle et al.	)	Examiner: Jay A. Morrison
	)	
Serial No.: 10/656,020	)	Group Art Unit: 2168
	)	
Filed: September 5, 2003	)	Appeal No.: _____
	)	
Title: PROJECT STRUCTURE	)	

REPLY BRIEF OF APPELLANTS

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with 37 CFR §41.41, Appellants hereby submit the Appellants' Reply Brief on Appeal from the final rejection in the above-identified application, as set forth in the Office Action dated April 16, 2008 and the Examiner's Answer dated December 10, 2008.

No fees are due at this time. However, should any fees be due, please charge any additional fees or credit any overpayment to Gates & Cooper LLP, Deposit Account No. 50-0494.

I. REAL PARTY IN INTEREST

The real party in interest is Autodesk, Inc. the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for the above-referenced patent application.

III. STATUS OF CLAIMS

Claims 1-30 are pending in the application.

Claims 1-9, 11-19 and 21-29 stand rejected under 35 U.S.C. §103(a) as being obvious in view of Bondy et al., U.S. Publication 2002/019219 (Bondy), Halpert et al., U.S. Publication 2004/0225958 (Halpert) and Fujieda, U.S. Publication 2002/0083082 (Fujieda).

Claims 10, 20 and 30 stand rejected under 35 U.S.C. 103(a) as being obvious in view of Bonday, Halpert, Fujieda and Rappaport et al., U.S. Patent 6,850,946 (Rappaport).

All of these rejections are being appealed.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been made subsequent to the final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The claim limitations and their support in the specification are set forth below:

Claim Limitation	Support in Specification
1. A computer-implemented method for defining a project in a computer graphics program comprising:	Page 6, lines 20-21; Page 17, lines 6-7; Fig. 6
(a) obtaining a project file in the computer graphics program comprising general information regarding the project;	Page 7, lines 10-12; Page 11, lines 7-9; Page 17, lines 8-9; Fig. 6 – 600.
(b) creating a directory structure in the computer graphics program for the project wherein:	Page 11, lines 3-4; Page 11, lines 20-22; Page 12, lines 1-2; Fig. 5; Fig. 6 – 602
(i) one or more project drawing files are organized into various	Page 5, lines 10-13; Page 7, lines 12-16; Page 11, lines 14-20; Page 12, lines 3-4; Page 17, lines 13-

folders by drawing file type of the one or more project drawing files;	18; FIG. 6 – 602
(ii) the one or more project drawing files are composed of either a building information model for the project or a report generated from the building information model; and	Page 10, lines 1-3; FIG. 4A – 402 and 404;
(iii) the one or more project drawing files are organized into the various folders based on the building information model or the report accordingly;	Page 10, lines 1-8; Page 11, lines 3-4; Page 11, lines 14-22; Page 17, lines 13-18; FIG. 6 – 602
(c) obtaining a companion file for each project drawing file, wherein each companion file provides information used to create the directory structure and comprises information to link each project drawing file to the project based on the building information model or the report; and	Page 5, lines 11-13; Page 7, lines 13-16; Page 11, lines 16-18; Page 18, lines 4-9; FIG. 6 – 604; Page 11, lines 3-4; Page 11, lines 20-22; Page 12, lines 1-2; Fig. 5; Fig. 6 – 602; Fig. 4C – 406B, 408B, 410B, and 412B
(d) displaying, in the computer graphics program on a display device, the one or more project drawing files in the various folders.	Page 8, lines 3-10; Fig. 1 – 102; Page 9, lines 2-3; Page 16, lines 15-23;
11. An apparatus for defining a project in a computer graphics program comprising:	Page 2, line 14; Page 6, lines 20-21; Page 17, lines 6-7; Fig. 6
(a) a computer having a memory;	Page 7, lines 20-22; Fig. 1 – 100 and 104
(b) an application executing on the computer, wherein the application is configured	Page 8, lines 3-10; Fig. 1 – 108

to:	
(i) obtain a project file comprising general information regarding the project;	Page 7, lines 10-12; Page 11, lines 7-9; Page 17, lines 8-9; Fig. 6 – 600.
(ii) create a directory structure for the project wherein:	Page 11, lines 3-4; Page 11, lines 20-22; Page 12, lines 1-2; Fig. 5; Fig. 6 – 602
(1) one or more project drawing files are organized into various folders by drawing file type of the one or more project drawing files;	Page 5, lines 10-13; Page 7, lines 12-16; Page 11, lines 14-20; Page 12, lines 3-4; Page 17, lines 13-18; FIG. 6 – 602
(2) the one or more project drawing files are composed of either a building information model for the project or a report generated from the building information model; and	Page 10, lines 1-3; FIG. 4A – 402 and 404;
(3) the one or more project drawing files are organized into the various folders based on the building information model or the report accordingly;	Page 10, lines 1-8; Page 11, lines 3-4; Page 11, lines 14-22; Page 17, lines 13-18; FIG. 6 – 602
(iii) obtain a companion file for each project drawing file, wherein each companion file provides information used to create the directory structure and comprises information to link each project drawing file to the	Page 5, lines 11-13; Page 7, lines 13-16; Page 11, lines 16-18; Page 18, lines 4-9; FIG. 6 – 604; Page 11, lines 3-4; Page 11, lines 20-22; Page 12, lines 1-2; Fig. 5; Fig. 6 – 602; Fig. 4C – 406B, 408B, 410B, and 412B

project based on the building information model or the report; and	
(iv) display, on a display device, the one or more project drawing files in the various folders.	Page 8, lines 3-10; Fig. 1 – 102; Page 9, lines 2-3; Page 16, lines 15-23;
21. An article of manufacture comprising a program storage medium readable by a computer and embodying one or more instructions executable by the computer to perform a method for defining a project in a computer graphics program, the method comprising:	Page 2, line 14; Page 26, lines 20-23; Page 6, lines 20-21; Page 17, lines 6-7; Fig. 6
(a) obtaining a project file comprising general information regarding the project;	Page 7, lines 10-12; Page 11, lines 7-9; Page 17, lines 8-9; Fig. 6 – 600.
(b) creating a directory structure for the project wherein:	Page 11, lines 3-4; Page 11, lines 20-22; Page 12, lines 1-2; Fig. 5; Fig. 6 – 602
(i) one or more project drawing files are organized into various folders by drawing file type of the one or more project drawing files;	Page 5, lines 10-13; Page 7, lines 12-16; Page 11, lines 14-20; Page 12, lines 3-4; Page 17, lines 13-18; FIG. 6 – 602
(ii) the one or more project drawing files are composed of either a building information model for the project or a report generated from the building information model; and	Page 10, lines 1-3; FIG. 4A – 402 and 404;
(iii) the one or more project drawing files are organized into the	Page 10, lines 1-8; Page 11, lines 3-4; Page 11, lines 14-22; Page 17, lines 13-18; FIG. 6 – 602

various folders based on the building information model or the report accordingly;	
(c) obtaining a companion file for each project drawing file, wherein each companion file provides information used to create the directory structure and comprises information to link each project drawing file to the project based on the building information model or the report; and	Page 5, lines 11-13; Page 7, lines 13-16; Page 11, lines 16-18; Page 18, lines 4-9; FIG. 6 – 604; Page 11, lines 3-4; Page 11, lines 20-22; Page 12, lines 1-2; Fig. 5; Fig. 6 – 602; Fig. 4C – 406B, 408B, 410B, and 412B
(d) displaying, in the computer graphics program on a display device, the one or more project drawing files in the various folders.	Page 8, lines 3-10; Fig. 1 – 102; Page 9, lines 2-3; Page 16, lines 15-23;

In view of the above, it may be noted that independent claims 1, 11, and 21 are generally directed to defining a project in a computer graphics program. More specifically, a project file is obtained that provides general information regarding a project. A directory structure is then created for the project. Project drawing files are organized into various folders of the directory structure by drawing file type. Further, the drawing files are composed of either a building information model component (for the project) or a report generated from the building information model. The organization into the various folders is further based on the model or report accordingly. A companion file for each project drawing file is obtained. Each companion file provides information used to create the directory structure that the files are organized in and also provides information to link each project drawing file to a particular project (based on the building information model or report). Lastly, the drawing files are displayed in the various folders within the graphics application.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-9, 11-19 and 21-29 are patentable under 35 U.S.C. §103(a) in view of Bondy et al., U.S. Publication 2002/019219 (Bondy), Halpert et al., U.S. Publication 2004/0225958

(Halpert) and Fujieda, U.S. Publication 2002/0083082 (Fujieda).

Whether claims 10, 20 and 30 are patentable under 35 U.S.C. 103(a) in view of Bondy, Halpert, Fujieda and Rappaport et al., U.S. Patent 6,850,946 (Rappaport).

VII. ARGUMENT

A. Claims 1-9, 11-19 and 21-29 Are Patentable Under 35 U.S.C. §103(a) in View of Bondy, Halpert and Fujieda.

1. Independent claims 1, 11, and 21

Appellant traverses the rejections for one or more of the following reasons:

(1) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest a computer graphics program;

(2) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest a project file in a computer graphics program;

(3) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest a drawing in a computer graphics program;

(4) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest a drawing file have a drawing file type;

(5) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest organizing a drawing file into a folder based on a drawing file type;

(6) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest a building information model or a report from a building information model;

(7) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest organizing drawing files into a folders based on a building information model or report from a building information model;

(8) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest a companion file for each drawing file; and

(9) Neither Bondy, Halpert, nor Fujieda teach, disclose or suggest a companion file for each drawing file that provides information to both (i) create a directory structure, and (2) to link each drawing file to a project based on the building information model or report.

Independent claims 1, 11, and 21 are generally directed to defining a project in a computer graphics program. More specifically, a project file is obtained that provides general information regarding a project. A directory structure is then created for the project. Project drawing files are organized into various folders of the directory structure by drawing file type. Further, the drawing files are composed of either a building information model component (for the project) or a report generated from the building information model. The organization into the various folders is further based on the model or report accordingly. A companion file for each project drawing file is obtained. Each companion file provides information used to create the directory structure that the files are organized in and also provides information to link each project drawing file to a particular project (based on the building information model or report). Lastly, the drawing files are displayed in the various folders within the graphics application.

The cited references do not teach nor suggest these various elements of Appellants' independent claims.

Appellants first note that the claimed invention is directed towards a computer graphics program. The preamble requires a computer graphics program and the first claim element obtains a project file in the computer graphics program. In rejecting this claim element, the Office Action relies on the abstract, background, and paragraph [0018] of Bondy. Appellants note that Bondy's abstract indicates that Bondy is directed towards printing a project of documents containing variable data and not a computer graphics program. In this regard, printing documents is not even remotely similar to a computer graphics program or drawings in a computer graphics program. Bondy's background further describes the process of printing fixed data and variable data in a "printing application". Again, a printing application is not a computer graphics program as claimed. Bondy's paragraph [0018] further describes a process for printing variable data document projects. In this regard, Bondy does not teach, describe, suggest, or remotely allude to a computer graphics program.

Appellants further note that the claims provide for project drawing files in the computer graphics program. Such claim limitations also provide that the drawing files are organized into various folders by the drawing file type of the drawing files. Such limitations further provide the context of the invention in that it relates to drawings and drawing files in a computer graphics program. In addition, such drawing files are organized based on the type of drawing file. In

rejecting this claim element, the Office Action refers to Bondy's "stored in folders" set forth in paragraph [0019]. Paragraph [0019] provides:

> [0019] Resources for the project, such as images, fonts, and graphics, are acquired in step 204. Also, in step 204, the resources for the project are stored in folders, i.e. directories, in accordance with the configuration file and tagged with appropriate metadata tags to identify the resources and associate the resources with the proper project and documents. In step 206, test data is acquired. Test data can be any set of data corresponding to expected variable data, such as a single representative record from a database to be used for variable data. In step 208, a counter is added to the file to provide a unique sequence number to each record of the project.

As can be seen from this text, the only mention of graphics is when it is a graphic for a document project. Thus, Bondy still fails to describe a computer graphics program. Further, a drawing file (i.e., a file containing a drawing) is not similar to a document that contains a font or graphic. Again, a drawing and computer graphics program provide a particular context that is neither hinted at or suggested in Bondy. In addition, the claimed drawing files are organized into folders by drawing file type. Nowhere in the cited text (or remainder of Bondy) is there a remote reference to organizing files (not to mention that they are drawing files) in any location based on the type of file (or type of drawing file) as claimed. Instead, Bondy describes storing resources for the project in folders based on a configuration file. In this regard, the configuration file is not a drawing file type. In addition, the claimed drawing files are stored in the folders based on their own drawing file type and not based on a separate configuration file.

In response to the above arguments, the Examiner's answer states that Bondy describes a system for printing documents which "could be considered a computer graphics program". The Answer continues and states that Fujieda teaches a computer graphics program and Halpert teaches a project file by reciting a Microsoft Project file.

Appellants respectfully disagree with and traverse such an assertion. Firstly, the assertion that a system for printing documents is equivalent to a computer graphics program is wholly without merit. The claim limitations consistently recite a "computer graphics program" as well as "drawing file types", "project drawing files", and a "building information model". All of such terms clearly refer to a computer graphics program and not a system for printing a project of documents as asserted in the Answer. In addition, Fujieda which provides for a CAD based application, cannot be combined with Bondy. There is simply no motivation to combine a system for printing a project of documents with a CAD program. In this regard, under KSR or otherwise, there is no reason to try

to combine them, nor is there a motivation in the art. Further yet, even if combined, the invention would not result. Lastly, for the project file aspect of the claims, the Answer relies on Halpert's Microsoft project. The claim limitations do not provide for merely having a single project file. Instead, as claimed, the project file is general information regarding the project and the remaining claim elements provide details relating to files, a directory structure, and the organization of both for the project. Halpert fails to teach such detailed claim elements.

The present claims continue to provide that the project drawing files are composed of either a building information model for the project or a report generated from the model. As can be seen throughout the text of the specification as filed, a building information model is an information model for a building (see paragraphs [0015]-[0017], [0038], [0040], etc.). Accordingly, the use of the term "building information model" in the claims provides a specific meaning and intent that can't merely be ignored. Under MPEP §2142 and 2143.03 "To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)." In this regard, the term "building" that modifies "information model" as used in the claims cannot merely be disregarded. Bondy's template has no relationship whatsoever to a building information model as claimed. Instead, Bondy provides that static portions of a print job are created as a template. Such a teaching is not even remotely similar to a building information model as claimed. The structure and use of the various folders and the drawing files in the particular folders provides functional advantages in the building industry (as further defined in the independent claims). Thus, to equate a template in a printing application with drawing files in a building information model is logically flawed.

In addition, the claims provide that the drawing files are stored/organized in the folders based on the building information model or report. In rejecting this claim element, the Office Action relies on Bondy's repository in paragraph [0020]. However, Bondy merely describes the creation of a markup of a page layout design that is imported into a repository. Bondy then provides that updates are stored in the repository. Such a teaching again fails to describe a claimed building information model. Further, the claimed limitations relating to the organization of drawing

files into folders based on an information model is completely ignored in a mere recitation of a repository that is used for a markup of a page layout design and updates of a static portion of an image. Again, there is not even a remote similarity between Bondy's teaching and the claimed invention with respect to such specifically and explicitly claimed elements.

The present claims then provide that a companion file is created for each project drawing file. In rejecting this claim element, the Office Action relies on Bondy's configuration file. Appellants note that Bondy's configuration file stores "the file structure, ID, and other project specific data" (see paragraph [0018]). Thus, Bondy's configuration file is a single file that is used to store all document project information. Accordingly, Bondy's configuration file is not created separately for each project file as claimed. In this regard, instead of creating a companion file for each project drawing file (as claimed), Bondy creates a single configuration file that is used on a project wide basis. Such a teaching does not and cannot teach the companion files for each project drawing file as claimed.

The present claims then provide that the companion file provides information used to create the directory structure. There is no mention or even remote suggestion in Bondy for creating a directory structure based on the configuration - not to mention creating a directory structure for the companion files that are created for each project drawing file as claimed.

The present claims further provide that each companion file has information to link each project drawing file to the project based on the building information model or the report. Again, a single configuration file that merely provides a file structure, ID, and other project specific data does not and cannot possibly teach a companion file that is used to link each and every project drawing file to a project wherein such a link is based on a building information model or report as claimed.

In response to such arguments, the Answer (page 16) again asserts relies on Bondy's configuration file. The Answer further asserts that Bondy's entire project could represent a single document which would read on the project drawing file for which the configuration file stores resources in folders. Appellants respectfully disagree with and traverse such an assertion. Firstly, as stated above, Bondy's document printing system is not equivalent to a project drawing file as claimed. Further, the Examiner's application to the claims is inconsistent. In rejecting the drawing files and drawing type, the Examiner asserts an equivalency to the resources. Thus, the Examiner

asserts multiple resources. Now, the examiner states that the entire project and not the resources are equivalent to the project drawing file. Such an interpretation is inconsistent and an improper application of the cited art to the claims. In accordance with the Examiner's earlier interpretation, each resource would have to have a companion file. Bondy does not have a configuration file for each resource. Alternatively, if Bondy's entire project is the claimed project drawing file, then the entire project would have to be organized into folders by the type of project – something that Bondy clearly fails to provide. Thus, no matter how one interprets Bondy's resources or project, it is impossible to equate Bondy to the present claims.

Appellants further note that the ability to link the file based on the building information model provides the ability to manage drawing files that have different meaning in different folders. Such a functional advantage is further illustrated in claims 6-9. These claims provide a more detailed context for the building information model aspect of the claims. In this regard, the folders and respective drawing files provide for elements, constructs, views, and sheets - all of which have specific identifiable meanings as set forth in the claims. In rejecting each of these claims, the Office Action relies on Halpert. However, Halpert is not in the building information industry and is such unrelated art that it cannot be applied to the present invention. In this regard, Halpert relates to publishing or displaying structured data at a website (see Abstract). The only mention of a project in Halpert relates to the Microsoft™ program Microsoft Project™. Thus, such a reliance in the Office Action to a project application when rejecting claims in a project having project drawing files is completely and entirely misplaced.

In response to the above arguments, the prior final Office Action first states that Appellants argues the preamble. Appellants respectfully disagree with and traverse such an assertion. While the preamble discloses that a project is defined, the actual claim limitations explained in the above arguments serve to perform such a defining of the project. Further, the actual claim limitations provide for the computer graphics program.

The final Office Action continues on page 9 and states that the contents of the files themselves are immaterial since the data is not made functional in the claims and is just treated as any other type of data. Appellants respectfully disagree with and traverse such an assertion. In this regard, the claim limitations absolutely provide functional limitations with respect to the content of

the files. For example, the project drawing files are organized into folders by drawing file type of the one or more project drawing files. Thus, rather than merely organizing files into random folders, the type of drawing – i.e., drawing file types are used to organize the files into particular folders. Further, while the project drawing files are made up of either a building information model or a report from such a model, the claim limitations explicitly provide that the files are also organized into the folders based on such a model or report. Further yet, each project drawing file is linked to a project based on such a model or report. Accordingly, contrary to that asserted in the final Office Action, it is impossible to separate the claim limitations and their functional aspects relating to both how the files are organized and how the files are linked to a project.

As stated above, Appellants reassert that nowhere in the cited text (or remainder of Bondy) is there a remote reference to organizing files (not to mention that they are drawing files) in any location based on the type of file (or type of drawing file) as claimed. Instead, Bondy describes storing resources for the project in folders based on a configuration file. In this regard, the configuration file is not a drawing file type. In addition, the claimed drawing files are stored in the folders based on their own drawing file type and not based on a separate configuration file. Instead, paragraph [0018] are relied upon to allegedly teach a directory structure where files are organized into various folders based on file content. Paragraph [0018] provides:

> [0018] FIG. 3 is a flowchart of the process for printing variable data document projects in accordance with the embodiment. The process can be divided into two distinct phases. The first phase is the creation phase (to the left of the dotted line) and the second phase is the production phase (to the right of the dotted line). Referring to FIGS. 1 and 2, each component communicates with operations management component 32 to permit operations management component 32 to maintain a real-time status of each project. For example, such communication can be in an HTTP compliant format using XML messaging in the manner described below. The process begins in step 200 in which a project is created by operations management component 32. In step 202, a file structure and directory structure for the project is set up in repository 140 in accordance with an assigned identification, such as a customer ID and a job ID. The ID is used as metadata to permit tracking and reporting of status and other variables. The file structure, ID and other project specific data can be stored as a configuration file for the project.

As can be seen from this text (and the remainder of Bondy), there is no reference nor teaching, implicit or explicit, regarding the organization of files into folders based on a file type whatsoever. Instead, a project is created and a file structure and directory structure for the project are set up in a repository. However, such a teaching does not even remotely refer to organizing files into a folder based on a file type (or more specifically a drawing file type).

In response to such arguments, the Examiner's Answer no relies on paragraph [0019] of Bondy and asserts:

> Here we see that the resources are defined by type such as images and fonts and stored in folders which shows that the type of the resource defines the folder in which it is stored. Here Bondy teaches that images are stored in a certain folder and fonts in another folder, which shows that the type of the resource defines where it is stored.

Paragraph [0019] provides:

> [0019] Resources for the project, such as images, fonts, and graphics, are acquired in step 204. Also, in step 204, the resources for the project are stored in folders, i.e. directories, in accordance with the configuration file and tagged with appropriate metadata tags to identify the resources and associate the resources with the proper project and documents. In step 206, test data is acquired. Test data can be any set of data corresponding to expected variable data, such as a single representative record from a database to be used for variable data. In step 208, a counter is added to the file to provide a unique sequence number to each record of the project.

As can clearly be seen, the Examiner's interpretation of such text is misplaced. Firstly, a resource is not the file itself as is presently claimed. Instead, Bondy's resource is a resource for a document. In addition, the resource is stored in a folder not based on the resource, but based on a configuration file. Nowhere in the text of Bondy does it state that the images and fonts are stored in a particular folder based on its type. Instead, it clearly states that the storage location is based on the configuration file. Further, contrary to that asserted in the Answer, such text does NOT state that images are stored in a certain folder and fonts in another folder. In fact, Appellants assert that the text shows that all of the resources (images, fonts, graphics, etc.) are all stored in the same folders/directories based on a configuration file. There is simply no description in Bondy that even remotely states that different folders are used for different types of resources. Thus, the Examiner's interpretation is simply wrong and misleading. Further, the Examiner acknowledges that the storage is in accordance with such a configuration file later in the answer (see Answer page 15, last paragraph and Answer page 16, first paragraph).

The claim limitations also provide that the drawing files are composed of either a building information model or report. Further, such a model or report is used when organizing the files into folders, and the companion file links the project drawing file to the model or report. No such building information model or report are even remotely alluded to in Bondy. Further, the final Office Action fails to address the arguments relating to such claim elements.

In response to such arguments, the Answer now relies on paragraph [0020] and states that such text describes a page layout design and a template which is created based on the design. The Answer further asserts that the layout design and template are the model upon which the project can be created and stored to correspond to the project and therefore reads on the claimed project drawing files that are a model or a report generated by the model. Again, as stated above, Bondy describes a system for printing a project of documents and not the ability to define a project as claimed. The claims explicitly provide: "the one or more project drawing files are composed of either a building information model for the project or a report generated from the building information model". To state that a page layout design and template constitute a business information model is disingenuous. There is simply no model set forth in Bondy that even remotely alludes to building information.

The Answer then asserts that the claims don't provide details regarding how a building information model is different from any other model and accordingly, Bondy's layout design can read on the claimed model. Appellants respectfully disagree. Based on such an analysis, the Examiner finds a convenient mechanism for simply disregarding the explicit claim language. The building information model is used in a computer graphics program with respect to drawing files. It can further clearly be seen by the dependent claims (e.g., claim 7) that the project and building information model relates to a building. Further, the specification further relates a building information model to a building (see e.g., para [0044]). Thus, Appellants are entitled to be their own lexicographer and the Patent Office cannot simply ignore the express claim limitations. Bondy does not and cannot read on any buildings or a building information model in any manner, because Bondy merely deals with printing documents.

In addition, the claim limitations explicitly provide that the companion file is used to "create the directory structure". Nowhere in Bondy, explicitly or implicitly, is there any teaching, hint, suggestion, or otherwise regarding the creation of a directory structure whatsoever. Further, the ability for a companion file to provide information that is used to create such a directory structure is completely and entirely lacking from Bondy. The final Office Action completely disregards and ignores the explicit claim limitations directed towards directory structure creation and merely glosses over the limitations stating "companion file with metadata tags to identify resources, paragraph

[0019], lines 1-8." Such a summary rejection is improper and fails to address each of the claim limitations. Accordingly, the final Office Action has failed to establish a prima facie case of nonobviousness.

In response to the above previously asserted arguments, prosecution was reopened and the Examiner acknowledged Bondy and Halpert's lack of teaching a computer graphics program. Instead, the Patent Office now relies on Fujieda. While Fujieda may disclose a computer graphics program, the claims have significant limitations all of which are lacking from Fujieda. Further, without disclosing a computer graphics program, Bondy and Halpert cannot possibly teach, disclose, or suggest a project file, directory structure, etc. in the computer graphics program as explicitly and expressly claimed. In this regard, it is improper and meritless to attempt to merely utilize a reference that discloses a computer graphics program and combine such a reference with an entirely separable and unrelated reference. The final Action asserts that they references can be combined by enabling the management of different types of CAD data and to provide the user the advantage of a single source for data management. However, Bondy is related to printing documents and variable data document printing - a concept not even remotely related to CAD data whatsoever or to a single source of data management. Thus, there is no rationale or reason to combine Bondy with Fujieda or with Halpert.

In further response to the above asserted arguments, the final Action and Answer again repeats the reliance on Bondy paragraphs [0018]-[0020]. For the reasons stated above, Appellants respectfully disagree with and reassert the above arguments.

The Action then asserts (i.e. on page 12) that the data contents are not made functional in the claims and are thus treated as non-functional descriptive material. As stated above, the claim limitations absolutely provide functional limitations with respect to the content of the files. For example, the project drawing files are organized into folders by drawing file type of the one or more project drawing files. Thus, rather than merely organizing files into random folders, the type of drawing – i.e., drawing file types are used to organize the files into particular folders. Further, while the project drawing files are made up of either a building information model or a report from such a model, the claim limitations explicitly provide that the files are also organized into the folders based on such a model or report. Further yet, each project drawing file is linked to a project based on such

a model or report. Accordingly, contrary to that asserted in the final Office Action, it is impossible to separate the claim limitations and their functional aspects relating to both how the files are organized and how the files are linked to a project.

Nonetheless, even if one removes the word "drawing" from the "files" it can clearly be seen that Bondy and the other cited references still fail to teach describe or suggest files that are organized into folders by file type of the project files, a model for a project or a report generated from the model, organizing files into folders based on the model or report, companion files for each project file, the companion file being used to create a directory structure and has information that links each project file to a project based on a model or report, and the ability to display project files in the various created folders. Again, regardless of whether one considers the language functional or not, the elements and limitations that are utilized in the claims serve to clearly and significantly distinguish the cited references from the present invention.

In view of the above, Appellants respectfully request allowance of the rejected claims.

2. Dependent claims 2, 12, and 22

These dependent claims provide that the general information regarding the project is selected from a group consisting of: a project name, a project number, a project level, a project division, a first default template for a new element, a second default template for a new construct, a third default template for a new view, and a fourth default template for a new sheet.

MPEP 2111.03 (and supporting case law) provides that the transitional phrase "consisting of" excludes any element, step, or ingredient not specified in the claim. In other words, the claim is closed to the inclusion of materials other than those recited.

In rejecting these claims, the prior final Office Action recites each of the above items in the group followed by a reference to "data, paragraph [019]" of Bondy.

Paragraph [0019] provides:

> [0019] Resources for the project, such as images, fonts, and graphics, are acquired in step 204. Also, in step 204, the resources for the project are stored in folders, i.e. directories, in accordance with the configuration file and tagged with appropriate metadata tags to identify the resources and associate the resources with the proper project and documents. In step 206, test data is acquired. Test data can be any set of data corresponding to expected variable data, such as a single representative record from a database to be used for variable data. In step 208, a counter is added to the file to provide a unique sequence number to each record of the project.

Appellants submit that such language in paragraph [0019] fails to teach a project level, a project division, a first default template for a new element, a second default template for a new construct, a third default template for a new view, and a fourth default template for a new sheet. Nowhere in Bondy's paragraph [0019] (or the remainder of Bondy) is there even a remote reference to such explicit claim limitations. Further, the Office Action fails to address each of the claim limitations with the necessary specificity to establish a prima facie case of nonobviousness.

In response to the above previously submitted arguments, the Office Action and Answer assert that the noted differences are only found in nonfunctional descriptive material and are not functionally involved in the steps recited.

Appellants respectfully disagree with and traverse such assertions. In this regard, the listing of information is used in the independent claims since it identifies and provides information regarding the project. The independent claims further link each project drawing file to the project. Such general information is used as part of such an association/linkage between the project drawing files and the project itself. Thus, the different types of general information provide a context and format for the project file itself. Such limitations clearly extend functional capabilities and further establish an association between the project file itself and a project (i.e., for a building information model). Thus, the claim language is functional and provides a functional advantage over that of the prior art.

In view of the above, Appellants respectfully request reversal of the rejections.

3. Dependent claims 3, 13, and 23

These dependent claims provide that the project drawing file is an XML document. In rejecting the claims, the prior final Office Action merely refers to Bondy paragraph [0018] which provides:

> [0018] FIG. 3 is a flowchart of the process for printing variable data document projects in accordance with the embodiment. The process can be divided into two distinct phases. The first phase is the creation phase (to the left of the dotted line) and the second phase is the production phase (to the right of the dotted line). Referring to FIGS. 1 and 2, each component communicates with operations management component 32 to permit operations management component 32 to maintain a real-time status of each project. For example, such communication can be in an HTTP compliant format using XML messaging in the manner described below. The process begins in step 200 in which a project is created by operations management component 32. In step 202, a file structure and directory structure

> for the project is set up in repository 140 in accordance with an assigned identification, such as a customer ID and a job ID. The ID is used as metadata to permit tracking and reporting of status and other variables. The file structure, ID and other project specific data can be stored as a configuration file for the project.

As can clearly be seen, rather than teaching a drawing file that is an XML document, paragraph [0018] merely provides that communications can occur via XML messaging. Such a teaching is not similar to nor does it teach, describe, or suggest that a project drawing file is an XML document.

In response to the above arguments, a subsequent Office Action agrees that Bondy failed to disclose the limitations and instead relies on Halpert paragraph [0104], lines 2-10 which provides:

> [0104] At this point the VisioDataClient will extract all of the necessary information from Visio for the dropped file. From this information it will generate ApXML data file that describes the project element hierarchy. The top-most project element will represent the drawing itself. The project elements under this will either represent all of the shapes on the layers selected to be converted, or it will represent each page in a multi-page drawing. Each element in the ApXML data file that represents a page or shape from the original drawing is marked in such a way that an association is created between the ActiveProject data model object and the original visio object that it represents. Therefore the data model objects "remember" where they came from. ApXML elements are marked with a unique identifier stored in a CustomProp tag. In the case of a page it is the page number, in the case of a shape it is the shape id. If the same Visio drawing file is modified and dropped on the Inbox again, ActiveProject uses these tags to recognize what has changed and update the web site accordingly. The system thus ensures that all changes made to the web site after the original publishing operation are maintained properly on subsequent republishes.

Such text provides that an ApXML file is generated that describes a project element hierarchy. Each element in the XML file may represent a page or shape is marked to establish an association between an object model and a visio object that the element represents.

Appellants note that the claimed project drawing file does not describe a project element hierarchy for another drawing but is an XML file itself. Further, the claimed XML project drawing file is organized into folders by drawing file type, is a building information model for a project or report, etc. However, Halpert's XML file merely describes a hierarchy for elements in a drawing - it is not for a project, it is not a report, and fails to include numerous limitations as presently claimed. Accordingly, it is impossible for Halpert's ApXML file to be equivalent to or even remotely suggest the claimed XML project drawing file.

In response to the above, the Examiner's Answer now relies on Halpert paragraph [0106] which again merely refers to an ApXML generated from a Visio drawing. There are various levels

with tags for faces and child faces. However, as stated above, there is nor organization into folders by drawing file type, nor is it for a building information model for a project or report, etc. There is simply no comparison and no capability to provide the specific structure set forth in the present claims.

In view of the above, Appellants respectfully request reversal of the rejections.

4. Dependent claims 4, 14, and 24

These dependent claims provide that the companion file is an XML file. The Office Action again merely relies on Bondy paragraph [0018]. Similar to claims 3, 13, and 23, rather than teaching a companion file that is an XML document, paragraph [0018] merely provides that communications can occur via XML messaging. Such a teaching is not similar to nor does it teach, describe, or suggest that a project drawing file is an XML document.

In response to such arguments, the new Office Action acknowledges Bondy's lack of teaching and instead relies on Halpert paragraph [0104], lines 2-10. The Answer further relies on Halpert paragraph [0106]. For the reasons set forth above with respect to claims 3, 13, and 23, Appellants disagree with and traverse such assertions.

In view of the above, Appellants respectfully request reversal of the rejections.

5. Dependent claims 5, 15, and 25

These dependent claims provide that the folders (in which the project drawing files are organized into) are: an elements folder for element type drawing files within the building information model, a constructs folder for construct type drawing files within the building information model, a views folder for view type drawing files for the report, and a sheets folder for sheet type drawing files for the report.

In rejecting these claims, the Office Action summarily refers to "directory structure, paragraph [0019]". As can be seen from paragraph [0019] (see above), the text does not even remotely refer to different file types, different folders, a building information model, or a report. The claim provides for specific claim limitations that cannot merely be ignored. The limitations

provided are specific to the building industry and provide a context and scope for the claims which was not available in the prior art. Further, the limitations provide advantages over the prior art.

In response to such arguments, the new Office Action asserts that such claim limitations are non functional and intended use limitations and are therefore not entitled to any patentable weight. Appellants respectfully disagree with such assertions. As stated above, such limitations provide a context for the claims, provides a specific number of folders (i.e., elements, constructs, views, and sheets for a total of four folders), establishes a specific corresponding folder for each type of specific drawing file, and further establishes different folders used for files within a building information model than those used for a report. All of such limitations are functional in nature and are clearly more than mere intended uses.

In response to the above arguments, the Examiner's Answer merely repeats that the language is non-functional and the "for" statement in each of the folder descriptions indicates intended use of the data inside of the directory. Appellants note that as explicitly claimed, the various folders still consist of each of the folders listed – an elements folder, a constructs folder, a views folder, and a sheets folder. The "for" statement identifies what type of files are stored in each folder. Thus, they are not merely "intended uses" but explicitly define what type of data is stored in each of the folders listed.

The Answer continues and states that Bondy discloses that "the resources for the project are stored in folder, i.e., directories" which the answer asserts is equivalent to the claimed directories in the claim language based on the non-functional aspects of the claim limitations. Appellants respectfully disagree. Even assuming that the language is non-functional, there are still a minimum of 4 different folders specifically identified in the claims, each of which are used for different types of drawing files. Merely reciting "resources are stored in folders" fails to teach, disclose, or suggest, fours specific different folders. Further, as stated above, resources are not drawing file types as claimed. Lastly, as stated above, the different folder types are functional because they provide an organizational structure that is used during the explicitly claimed creation of the directory structure step.

In view of the above, Appellants respectfully request reversal of the rejections.

6. Dependent claims 6, 16, and 26

These dependent claims depend on claims 5, 15, and 25 and provide that an element type of a drawing file is a set of geometry that is repeated throughout a project. In rejecting this claim element, the Office Action relies on Halpert paragraph [0084]. Paragraph [0084] does not even remotely allude to a set of geometry that is repeated in a drawing file. Instead, paragraph [0084] describes the ability to import any document that has structured data into a website. Such a teaching is not relevant whatsoever to the present claims.

Again, claims 6, 16, and 26 (in combination with claims 5, 15, and 25) are specifically directed towards the building industry and a CAD application having elements, constructs, views, and sheets. Such claim limitations are not even remotely alluded to in any of the cited references. In this regard, the reliance on both Bondy and Halpert is totally misplaced and illogical.

In responding to such arguments, the final Office Action asserts that Halpert discloses that a structure can be imported into a matching structure (paragraph [0084]). However, paragraph [0084] does not even mention geometry nor geometry that can be repeated in a project. Further, since the claims disclose a project in a computer graphics application, the ability to repeat a set of geometry in such a project clearly has functional aspects. By ignoring the computer graphics and project aspects of the claims, the final Office Action is disregarding the limitations that present useful and functional material.

A subsequent Office Action and the Answer repeats the prior arguments and further asserts that geometry is simply data since the functionality of the repeating of these elements is not clear, and the importing or repeating of a random element is taught by Halpert. Appellants again disagree and traverse such assertions. In reality, what the Examiner is asserting is that there is no functional difference between repeating a set of geometry throughout a project with repeating an element that has no geometric features. The only possible rationale that can lead to such a conclusion is to completely ignore the context of computer graphics applications and drawings. Again, the claims explicitly relate to and provide for project drawing files and a directory structure for a project in a computer graphics application. An element of a website may not have geometric features, may merely comprise text, and does not have the various specific elements, constructs, views, and sheets as is explicitly required in the present claims (in that they depend on dependent claims 5, 15, and 25).

The Examiner's analysis is not only improper and meritless but is illogical. It is clear that repeating a set of geometry in a project is not even remotely similar to repeating an element used in a website. There are not only functional differences but such a comparison cannot even be made in the first place because the differences are so clear and unambiguous.

In view of the above, Appellants respectfully request reversal of the rejection of these dependent claims.

7. Dependent claims 7, 17, and 27

Claim 7 further depends on claims 5, 15, and 25 and provides that the construct type drawing file is an identification of geometry and data for a particular level/wing and category of the project and one or more elements. Again, the level/wing aspect of the claims specifically relates to the building industry. In rejecting this claim, the Office Action relies on Bondy paragraph [0030] that states that a component tag is a descriptor identifying a type of information being sent. Appellants do not understand the relevance of such a recitation to the present claims. Such a statement does not disclose geometry, a level/wing, a category of a project, or elements, as claimed.

Again, claims 7, 17, and 27 (in combination with claims 5, 15, and 25) are specifically directed towards the building industry and a CAD application having elements, constructs, views, and sheets. Such claim limitations are not even remotely alluded to in any of the cited references. In this regard, the reliance on both Bondy and Halpert is totally misplaced and illogical.

In response to such arguments, the final Office Action asserts that the claim limitations are non-functional and descriptive and therefore no need exists to separately address these claim limitations beyond that of the rejections of independent claim 1. Appellants respectfully disagree with and traverse such an assertion. All of the claim limitations provide various functional limitations in that they specify the types of files that can be organized into particular folders and how to organize such files. Further, viewed in conjunction with the independent claims, the limitations explain and provide functional capabilities that are used by and in the independent claims. For example, the claim limitations in claims 7, 17, and 27 provide that the companion file provides information used to create a directory structure that contains a constructs folder and further provides that a construct type drawing file is stored in such a folder. Further, the construct file

identifies geometry and data for a particular level/wing and category of the project along with one or more elements. Such specific claim limitations go well beyond providing mere descriptive non-functional material.

In response to the above arguments, the prior Office Action and Answer agrees with Bondy's lack of teaching and again merely reasserts that the limitations are directed to nonfunctional descriptive material. Appellants reassert the arguments set forth above and submit that the Examiner's rejection is in error and fails to establish a prima facie case of unpatentability.

In view of the above, Appellants respectfully request reversal of the rejections of these dependent claims.

8. Dependent claims 8, 18, and 28

Claims 8, 18, and 28 further depend on claims 5, 15, and 25 and provide that the view type drawing file automatically assembles constructs to represent a portion of a project that has been selected based upon user specified data. In rejecting this claim, the Office Action relies on Halpert paragraph [0092]. This paragraph describes the steps that occur when a file is dropped onto an Inbox. Again, the relevance of such a description with respect to the present claims is unknown.

Again, claims 8, 18, and 28 (in combination with claims 5, 15, and 25) are specifically directed towards the building industry and a CAD application having elements, constructs, views, and sheets. Such claim limitations are not even remotely alluded to in any of the cited references. In this regard, the reliance on both Bondy and Halpert is totally misplaced and illogical.

In response to these prior arguments, the final Office Action asserts that the claim can be interpreted as directed towards most any type of data and therefore Halpert discloses the limitation. Appellants respectfully disagree with and traverse such an assertion. In this regard, the claims provide a functional limitation in that the view type drawing file functionally and automatically assembles appropriate constructs to represent a portion of a project that has been selected based upon user specified data. There is not even a remote possibility that such claim language merely depicts non-functional descriptive material. Not only is a portion of a project selected based upon user specified data, but the view type drawing file automatically assembles appropriate constructs to represent such a portion. Thus, there are two separate functional aspects in these claims. Nowhere

in paragraph [0092] nor the remainder of Halpert is there a capability to select a portion of a project based on user specified data. In this regard, what occurs when a file is dropped onto an Inbox does not remotely reflect either the selection of a portion of a project nor a selection based upon user specified data. In fact, such a description does not suggest the selection of a portion of anything.

In response to the above previously submitted arguments, a subsequent Office Action and the Answer merely repeats the prior rejections. Accordingly, Appellants reassert that arguments set forth above.

In view of the above, Appellants respectfully request reversal of the rejection of these claims.

9. Dependent claims 9, 19, and 29

Claims 9, 19, and 29 provide that the sheet type drawing file has one or more views and represents a printed/plotted document. In rejecting these claims, the Office Action relies on Bondy paragraph [0039] which describes the formatting of personalized catalogs for printing. However, there is no teaching or suggestion in Bondy relating to views for a drawing sheet as claimed.

Again, claims 9, 19, and 29 (in combination with claims 5, 15, and 25) are specifically directed towards the building industry and a CAD application having elements, constructs, views, and sheets. Such claim limitations are not even remotely alluded to in any of the cited references. In this regard, the reliance on both Bondy and Halpert is totally misplaced and illogical.

The final Office Action and Answer fails to address the arguments above and merely reasserts the same rejections. Accordingly, Appellants assume that the Examiner has acquiesced and agrees with the above arguments. Thus, Appellants respectfully request allowance of these claims.

B. Claims 10, 20 and 30 are Patentable Under 35 U.S.C. 103(a) in View of Bondy, Halpert, Fujieda and Rappaport

These dependent claims provide that the companion file is obtained by defining a user definable category and value for project information, followed by storing the user definable category and value in the companion file.

In response to prior arguments, multiple prior actions acknowledge Bondy's lack of teaching and relies on Harper paragraph [0081]. Further, the Office Action asserts that such steps:

…would have given those skilled in the art the tools to customize project information. This gives the user the advantage of having control over how a project is defined.

Paragraph [0081] provides:

> [0081] The system uses a meta-design that represents the hierarchy and interdependencies of the structure defined by the "original" data. A significant feature of the invention is that while the ApXML files may not be passed directly to the website, they define a meta-design object model. The meta-design object model, in turn, defines the website structure and the events that modify the website. The meta-design object model itself is persistent, and, among other advantages, enables the provision of back-references and "reconstructability" of the object model. Further information regarding such meta-designs can be found in Framework Technologies Corporation's U.S. Pat. No. 5,664,180, the disclosure of which is incorporated herein by reference. Within the meta-design, each aspect can be decomposed into hierarchical project elements and sub-project elements. Each project element may have associated properties and other information items such as files, URLs, or database queries. The meta-design also describes the graphic look for each project element, which could be a thumbnail picture or a hotspot over a background picture. Each element in the meta-design has a corresponding tag in ApXML. For example, the tag `FACE` is used to define a project element in a specific aspect. Other ApXML tags describe properties of each face of a project element. For example, ApXML uses the property tag `LOOKGRAPHIC` to set the graphic representation for that face. It uses the property tag `INFOITEM` to describe any associated information item files with that face. `INFOITEM` itself has sub-properties that describe the specifics of the information item, such as its name, description, document type, and file path. ApXML is used to describe hierarchical information.

As can be seen, such text refers to the use of a meta-design where files define a meta-design object model which in turn, defines a website structure and the events that modify the website. The object model is broken down into hierarchical project elements and sub-project elements. Each project element may have associated properties and other information such as filed, URLs, or database queries.

However, what is notoriously lacking from such a description is the ability to define a user-definable category and value for project information followed by the storage of such a category and value in a companion file. Again, the independent claims provide that the companion file exists for each drawing file and is used to create a directory structure and has information to link each drawing file to a project based on the building information model or the report. Harper fails to teach, describe, or suggest, a user definable category. Harper further fails to teach, describe, or suggest, explicitly or implicitly the storage of such a user definable category and value in a companion file that exists for each drawing file.

Appellants further appreciate the Examiner's assertion in the Office Action that such steps would provide tools to customize project information and thereby provide a user advantages.

However, without any prior art citations that support such steps, the Office Action is relying on impermissible hindsight. In this regard, there is nothing in the cited prior art that even remotely hints to the limitations set forth in these dependent claims.

In response to the above previously asserted arguments, the most recent Office Action as well as the Answer now admits that neither Bondy, Halpert, nor Fujieda indicate "user defined" and instead relies on Rappaport col. 6, lines 50-52. Col. 6, lines 34-57 provide:

> Obstructions/partitions are classified into categories. The user may define different categories of obstructions/partitions. A category is defined by a textual description (e.g., "External Brick Walls"), a vertical height (i.e., how tall is the wall), a color (to quickly distinguish it from entities belonging to other categories while viewing the drawing), and electromagnetic properties (discussed in further detail later). The category to which a given drawing entity (where an entity is either a line or polygon) has been assigned defines the type of obstruction/partition it represents. For example, if a given line has been assigned to the user-defined category of "Sheetrock Walls," then it shares the characteristics given by the user to all other entities within that category throughout the entire database drawing. The process of either creating new entities or changing the category to which the entity belongs is a simple point-and-click process using a mouse or other positioning device, by linking entities to a particular user defined category of partitions. The preferred embodiment allows the physical, electrical, and aesthetic characteristics of entities of the same category to be individually or collectively edited. Category designation is carried out by assigning a particular numerical value to the field of each entity, wherein the field is specified as part of the drawing database.

As can clearly be seen, Rappaport's categories are textual descriptions for an obstruction/partition in a drawing. However, these dependent claims provide limitations relating to obtaining a companion file for each drawing file that is used to create a directory structure and information to link each project drawing file to a project based on a building information model or report. As part of obtaining the companion file, a user definable category and value are defined for project information. The user definable category and value are stored in the companion file. Rappaport's user defined category is merely a category for an obstruction/partition. It is not for project information and is not stored in a companion file that is used as described above. The Office Action is attempting to take portions of references that deal with unrelated concepts and combining them in a manner that will not only provide inoperable results, but would not result in the claimed invention. In this regard, even if one were to utilize Rappaport's user defined categories, such categories are not for a project and hence would not result in the claimed companion file containing the required category and value as claimed.

In view of the above, Appellants respectfully request reversal of the rejections.

C. Conclusion

In light of the above arguments, Appellants respectfully submit that the cited references do not anticipate nor render obvious the claimed invention. More specifically, Appellants' claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103. As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

Respectfully submitted,

GATES & COOPER LLP

Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: February 10, 2009

By: /Jason S. Feldmar/
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/bjs

G&C 30566.255-US-U1

CLAIMS APPENDIX

1. A computer-implemented method for defining a project in a computer graphics program comprising:

(a) obtaining a project file in the computer graphics program comprising general information regarding the project;

(b) creating a directory structure in the computer graphics program for the project wherein:

(i) one or more project drawing files are organized into various folders by drawing file type of the one or more project drawing files;

(ii) the one or more project drawing files are composed of either a building information model for the project or a report generated from the building information model; and

(iii) the one or more project drawing files are organized into the various folders based on the building information model or the report accordingly;

(c) obtaining a companion file for each project drawing file, wherein each companion file provides information used to create the directory structure and comprises information to link each project drawing file to the project based on the building information model or the report; and

(d) displaying, in the computer graphics program on a display device, the one or more project drawing files in the various folders.

2. The method of claim 1, wherein the general information is selected from a group consisting of:

a project name;

a project number;

a project level;

a project division;

a first default template for a new element;

a second default template for a new construct;

a third default template for a new view; and

a fourth default template for a new sheet.

3. The method of claim 1, wherein the project drawing file comprises an extensible markup language (XML) document.

4. The method of claim 1, wherein the companion file comprises an extensible markup language (XML) file.

5. The method of claim 1, wherein the various folders comprise:
an elements folder for element type drawing files within the building information model;
a constructs folder for construct type drawing files within the building information model;
a views folder for view type drawing files for the report; and
a sheets folder for sheet type drawing files for the report.

6. The method of claim 5, wherein the element type drawing file comprises a set of geometry, wherein the set of geometry is repeated one or more times throughout a project.

7. The method of claim 5, wherein the construct type drawing file comprises:
an identification of geometry and data for a particular level/wing and category of the project; and
one or more elements.

8. The method of claim 5, wherein the view type drawing file automatically assembles appropriate constructs to represent a portion of a project that has been selected based upon user specified data.

9. The method of claim 5, wherein the sheet type drawing file comprises one or more views and represents a printed/plotted document.

10. The method of claim 1, wherein the obtaining a companion file further comprises:

defining a user definable category and value for project information;

storing said user definable category and value in the companion file.

11. An apparatus for defining a project in a computer graphics program comprising:

(a) a computer having a memory;

(b) an application executing on the computer, wherein the application is configured to:

(i) obtain a project file comprising general information regarding the project;

(ii) create a directory structure for the project wherein:

(1) one or more project drawing files are organized into various folders by drawing file type of the one or more project drawing files;

(2) the one or more project drawing files are composed of either a building information model for the project or a report generated from the building information model; and

(3) the one or more project drawing files are organized into the various folders based on the building information model or the report accordingly;

(iii) obtain a companion file for each project drawing file, wherein each companion file provides information used to create the directory structure and comprises information to link each project drawing file to the project based on the building information model or the report; and

(iv) display, on a display device, the one or more project drawing files in the various folders.

12. The apparatus of claim 11, wherein the general information is selected from a group consisting of:

a project name;

a project number;

a project level;

a project division;

a first default template for a new element;

a second default template for a new construct;

a third default template for a new view; and

a fourth default template for a new sheet.

13. The apparatus of claim 11, wherein the project file comprises an extensible markup language (XML) document.

14. The apparatus of claim 11, wherein the companion file comprises an extensible markup language (XML) file.

15. The apparatus of claim 11, wherein the various folders comprise:

an elements folder for element type drawing files within the building information model;

a constructs folder for construct type drawing files within the building information model;

a views folder for view type drawing files for the report; and

a sheets folder for sheet type drawing files for the report.

16. The apparatus of claim 15, wherein the element type drawing file comprises a set of geometry, wherein the set of geometry is repeated one or more times throughout a project.

17. The apparatus of claim 15, wherein the construct type drawing file comprises:

an identification of geometry and data for a particular level/wing and category of the project; and

one or more elements.

18. The apparatus of claim 15, wherein the view type drawing file automatically assembles appropriate constructs to represent a portion of a project that has been selected based upon user specified data.

19. The apparatus of claim 15, wherein the sheet type drawing file comprises one or more views and represents a printed/plotted document.

20. The apparatus of claim 11, wherein the application is configured to obtain the companion file by:

defining a user definable category and value for project information; and

storing said user definable category and value in the companion file.

21. An article of manufacture comprising a program storage medium readable by a computer and embodying one or more instructions executable by the computer to perform a method for defining a project in a computer graphics program, the method comprising:

(a) obtaining a project file comprising general information regarding the project;

(b) creating a directory structure for the project wherein:

(i) one or more project drawing files are organized into various folders by drawing file type of the one or more project drawing files;

(ii) the one or more project drawing files are composed of either a building information model for the project or a report generated from the building information model; and

(iii) the one or more project drawing files are organized into the various folders based on the building information model or the report accordingly;

(c) obtaining a companion file for each project drawing file, wherein each companion file provides information used to create the directory structure and comprises information to link each project drawing file to the project based on the building information model or the report; and

(d) displaying, in the computer graphics program on a display device, the one or more project drawing files in the various folders.

22. The article of manufacture of claim 21, wherein the general information is selected from a group consisting of:

a project name;

a project number;

a project level;

a project division;

a first default template for a new element;

a second default template for a new construct;

a third default template for a new view; and

a fourth default template for a new sheet.

23. The article of manufacture of claim 21, wherein the project file comprises an extensible markup language (XML) document.

24. The article of manufacture of claim 21, wherein the companion file comprises an extensible markup language (XML) file.

25. The article of manufacture of claim 21, wherein the various folders comprise:

an elements folder for element type drawing files within the building information model;

a constructs folder for construct type drawing files within the building information model;

a views folder for view type drawing files for the report; and

a sheets folder for sheet type drawing files for the report.

26. The article of manufacture of claim 25, wherein the element type drawing file comprises a set of geometry, wherein the set of geometry is repeated one or more times throughout a project.

27. The article of manufacture of claim 25, wherein the construct type drawing file comprises:

an identification of geometry and data for a particular level/wing and category of the project; and

one or more elements.

28. The article of manufacture of claim 25, wherein the view type drawing file automatically assembles appropriate constructs to represent a portion of a project that has been selected based upon user specified data.

29. The article of manufacture of claim 25, wherein the sheet type drawing file comprises one or more views and represents a printed/plotted document.

30. The article of manufacture of claim 21, wherein the method for obtaining a companion file further comprises:

defining a user definable category and value for project information; and

storing said user definable category and value in the companion file.

EVIDENCE APPENDIX

None.

RELATED PROCEDDINGS APPENDIX

None.